IBM Informix® Dynamic Server™ (IDS)
IDS Problem Determination Tutorial Series

Introduction to Problem Determination

Table of Contents

Introduction to Problem Determination

About this tutorial

Introduction

This tutorial provides an introduction to methodologies on effectively diagnosing problems on IDS, researching known IDS product defects and identifying what information is most useful to the Informix Technical Support staff. By guiding you thru examples of commonly encountered problems, you can start to become comfortable in dealing with any problem you encounter.

This tutorial is intended for all Database Administrators and System Administrators who use Informix Dynamic Server (IDS) across all platforms. It is assumed that you are comfortable with the workings of IDS and your platform's operating system tools.

Setup

To be able to follow along the examples of this tutorial, you should have IDS Family Version 7 or Version 9 installed along with the 'stores' database created.

Tutorial Conventions Used

When a tool or utility is first mentioned it will be shown in **bold** text.

All command statements and their output will be shown in a `monospaced` font.

Some examples will show specific command options which may change over time, which will always be documented in IDS documentation.

About the author

Mark Jalkiewicz has been a member of the Informix/IBM Regency Support Organization since 1995. He is certified in Informix Systems Administration and Informix Performance & Tuning. Mark has been assigned to the Informix/SAP development support staff for the last past 8 years. In this role, he has provided SAP and its R3 user community with expertise and solutions to database problems of varying degrees. He has authored white papers within the SAP organization on topics related to Backup and Restore, and has conducted internal seminars at worldwide SAP locations on IDS Administration.

You can reach Mark Jalkiewicz by locating his email address in the IBM Global Directory at http://www.ibm.com/contact/employees/us

Section 1 Problem Types & Utilities

Section 1.1 Introduction

Lets begin by briefly discussing several common types of problems and the various tools and utilities available to help assist in the resolution process. Knowing how to categorize a problem can help eliminate untimely delays such as contacting hardware vendor support instead of Informix Technical Support and vice-versa.

Section 1.2 Common types of problems

Regardless of whether a simple or complex database environment is used, the possibility that problems will be encountered can never be eliminated. Most problems or errors that occur fall under one of the following categories:

Operating system specific
Hardware or I/O specific
Application/SQL specific
IDS Instance specific

Problems resulting from these categories can be returned via a client application (abend or slow running transaction) , or by the database server (crash or poor performance).

Section 1.3 Operating system (OS) specific problems

OS specific problems are also referred to as system problems, as they have nothing to do with the database other than making the ability to connect to it or to retrieve data from it impossible and/or difficult to do. Occasionally you may discover in these situations, that not only do database commands & utilities fail to execute, OS commands such as **ls, mv, cp, dd** also fail to work. Immediate responses to these problems include contacting the system administrators (if they are unaware of the situation) to perform analysis. They may decide to stop and restart the machine depending on the messages in the OS system log. Should this fail to resolve or identify the problem, then opening a support case with the operating system vendor will become necessary.

Section 1.4 Hardware or I/O specific problems

Hardware problems or disk I/O problems very often mimic database malfunction and/or performance problems. Typically, the IDS system administrator receives notice from a user complaining their application failed due to a page fault error, or a query that ran in two minutes yesterday runs for over two hours today. The first reaction to these problems should be to rule out possible hardware problems. In the event of data/index page errors, root cause can or may be attributed to problems such as an IDS raw device

partition being cross-mounted with another raw device partition, disk controller problem, or in the case of poor performance, disk contention. It's also worth mentioning here that application and server crashes can also be the result of physical CPU errors.

The first common reaction to these types of problems is to contact IDS Technical Support with the belief that the IDS server is at fault; this is the proper thing to do. The technical support staff will instruct you to check for these types of errors, or may assist you in the determination by requesting you make available IDS diagnostic files. (This is covered in more detail in Section 3: IDS Diagnostics Collection).

Section 1.5 Application/SQL specific problems

Application problems include poor performance, erroneous results and program abends. SQL errors are transactional errors, where SELECT, INSERT, DELETE, or UPDATE statements result in problems. The main concern here is to differentiate between the two, so as to not incorporate blame on the IDS server when programming logic in the application may be to blame.

With respect to SQL specific problems, please refer to Section 4: IDS Diagnostics Collection.

Section 1.6 IDS instance specific problems

When the IDS instance starts to experience problems, the instance may either record assertion failure messages in its respective *message log*, and/or perhaps shutdown (crash). Almost all the time in either case the assertion failure will also result in the creation of an "af.xxx" file, commonly known as the 'af file'. This file will contain essential information, such as a stack trace of the offending thread that caused the server to assert. This information can then be used to map against known Informix PTS defects, or it can be forwarded to Informix Technical Support for further analysis. More information on af.xxx files is discussed in Section 34: IDS Diagnostics Collection.

Section 1.7 Conclusion

Properly identifying the type of problem that you are experiencing will help eliminate untimely delays in the resolution process. The common types of problems that are commonly experienced are operating system specific, hardware and/or I/O specific, application/SQL specific and IDS instance specific. Each type has its own set of utilities and/or methodologies in helping isolate root cause.

Section 2 Problem Description

Section 2.1 Introduction

The first step in the problem resolution process is to identify:

a. The problem and its symptoms.
b. The business impact of the problem.

Section 2.2 What is the problem and its symptoms?

Problems are first reported as the result of an error message being written into the IDS online message log, or perhaps a user complaining about a long running job, or that a client application continues to return a specific SQL error.

Section 2.3 A problem situation: I can't backup up my logical logs!

The DBA reports to you that they cannot backup the logical logs on the productive system, a *"write error no 5"* is associated with the failure. The DBA is very apprehensive and informs that if this problem is not resolved quickly, the instance will hang during productive hours. Worse yet, should the instance crash and a restore become necessary, there will be loss of data as the logical logs will have been lost.

Section 2.4 What is the business impact of this problem?

The first step in problem analysis is determining what impact a problem has on the ability to conduct business - this can also be equated as severity. Presently, you are able to conduct business as normal; however continuation of this problem will have an adverse effect, as the instance may hang or worse yet experience loss of data. Clearly, the log backup failure here is critical and from this point on, this problem will be characterized as critical/severe to all individuals directly or indirectly involved in the resolution process. One very important aspect to learn is: the people in position to help the most may mistakenly associate your problem as just a nuisance (low severity) if the proper level of severity is not communicated.

Section 2.5 Accurately describing a problem

Let's perform a review of the log backup problem by defining three components: the problem symptom, the problem reproduction and the problem description.

Problem Symptom: (What is the error?)
Logbackup failed - function write to tape failed code -1 errno 5

Problem Reproduction: (Is this problem reproducible? How & when does it occur?)
Reproduces very easily
When the DBA attempts to backup the logical logs

Problem Description:
A logical log backup process fails with an i/o error number 5

With answering the above three questions, you are able to describe the problem in a concise manner. This approach of short and to the point helps eliminate the possibility of needless information being added that can later cause confusion of the actual problem.

Section 2.6 Environment of a problem

It is very important to determine the environment where the problem originates. Useful required environment information is:

IDS Version
CSDK Version (if applicable)
Hardware Manufacture
OS and OS version
Latest Patch Rev of OS

Is the database server a dedicated server?
Is the disk subsystem where the database is stored accessed locally, or NFS mounted?
Is the disk subsystem on individual disks, or on a RAID array?
Are the client applications running locally or remotely?
Are the tape devices mounted locally or remotely?

Ask yourself if the application is running in a supported configuration?
It is possible that your configuration of IDS and OS are not supported, or perhaps your 3rd party application (SAP, PeopleSoft) is not certified for your combination of OS and IDS version. Remember that certification means that the product combinations have been tested.

Section 2.7 What has changed?

Problems that occur may happen immediately after a major hardware/software upgrade, or configuration change. One aspect often overlooked is failure to determine recent environment changes. It is prudent to always check for such changes, as the 'change' itself may be primarily responsible for the problem. Returning to our log backup failure problem, this is an event handled primarily by the i/o sub-system. The i/o system calls will first attempt to open the file or device; so, it's reasonable to conclude that a system

call resulted in this error. One task would be to identity what **resource** (disk/tape) the i/o process was trying to access when the failure occurred. If the process failed while attempting to access a tape device, it would be advisable to investigate if the tape device(s) or drivers were recently changed.

Section 2.8 Conclusion

Problem description may be simple, and then again it may be difficult depending on the environment complexity. Whatever the degree, the same questions need to asked and answered in order to accurately describe the situation. It's important to know the business impact, the reproduction criteria, the complete environment, all recent environment changes, and finally defining a concise accurate problem description.

Section 3 IDS Diagnostic Data Collection

Section 3.1 Introduction

As you might imagine, the IDS environment can be quite complex, and a large number of error codes and corresponding error messages are available. IDS provides mechanisms that can help pinpoint the root cause of a problem when an error situation occurs.

Section 3.2 The IDS message Log and location

The IDS server message log is an operating-system file. Its main purpose in life is to log all messages related to the server such as checkpoints, archives, parameter changes etc. It is also where information related to problem events is logged.

It is recommended that you monitor the message log regularly to ensure that processing is proceeding normally. If the database server experiences a failure, the message log serves as an audit trail for retracing the events that later develop into an unanticipated problem. Often, the database server provides the exact nature of the problem and the suggested corrective action in the message log.

The location of the IDS Message Log is defined by the online configuration file (ONCONFIG) parameter 'MSGPATH'.

A quick way to see the last 20 recorded lines to the messages files in with the command **`onstat –m`**

```
onstat –m

Informix Dynamic Server Version 7.31.UD2X7  -- Quiescent
(CKPT REQ) -- Up 1 days 00:51:48 -- 18224 Kbytes

15:01:42  Checkpoint Completed:  duration was 0 seconds.
15:01:42  Checkpoint loguniq 1, logpos 0x515018
15:06:42  Checkpoint Completed:  duration was 0 seconds.
15:06:42  Checkpoint loguniq 1, logpos 0x516018
……..
……..
15:66:42  Checkpoint loguniq 1, logpos 0x516018
```

Section 3.3 Assertion failure detected - 'af.xxx' files

As previously mentioned, when a server error is encountered, an af.xxx file will be created in the location provided by the configuration file parameter DUMPDIR in addition to the assert failure message being written to the online message log.

The figure below shows the form that assertion failures take in the message log.

Assert Failed: *Short description of what failed.*
Who: *Description of user/session/thread running at the time.*
Result: *Sate of the affected database server entity.*
Action: *What action the database administrator should take.*
See Also: *file(s) containing additional diagnostics.*

The See Also:line contains one or more of the following filenames:

af.xxx
shmem.xxx
gcore.xxx
/pathname/core

In all cases, xxx will be a hexadecimal number common to all files associated with the assertion failures of a single thread. The files af.xxx, shmem.xxx and gore.xxx are in the directory specified by the ONCONFIG parameter DUMPDIR.

The file af.xxx contains a copy of the assertion failure message that was sent to the message log , as well as the contents of the current, relevant structures and data buffers. The information will usually include the tablespace where the error was detected, with the Action the administrator should take, such as run oncheck –cD.

The file shmem.xxx contains a complete copy of the database server shared memory at the time of the assertion failure, but only if the ONCONFIG parameter DUMPSHEM is set to 1. This file is useful to Informix Technical Support when the af.xxx file does not contain the desired diagnostic information needed to determine root cause.

On UNIX, gcore.xxx contains a core dump of the database server virtual process on which the thread was running at the time , but only if the ONCONFIG parameter DUMPGCORE is set to 1and your operating system supports the gcore utility.

Section 3.4 IDS `finderr` utility

The `finderr` utility when invoked from a UNIX command line will echo back the error message and possible corrective recommendation.

Using our log backup failure as an example, we see that the error is returned by a client application, in this case the ontape utility. We can start to investigate a little further by invoking this utility. The usage is in the form **finderr** <*error no*>.

```
finderr 5

-5      I/O error.

An operating-system error code with the meaning shown was
unexpectedly returned to the database server. Look for other
```



```
operating-system error messages that might give more information.
In particular, try to determine what device produced the error
and whether that device held the database or some other kind of
file.
```

Examination of the above text reveals the origin of the error (operating system), and possible causes for the error (device or file problem). All operating system error codes on UNIX systems can be seen in /usr/include/sys/errno.h

Section 3.5 Other Informix Utilities

IDS comes with the following utilities to assist you in performing diagnostics tasks.

`oncheck:`
This utility is used for checking specified disk structures for inconsistencies , repairing corrupted indexes , and displaying information on disk structures. Disk structures would be tables, indexes, system catalog information , chunks and dbspaces.

`onstat:`
This utility is used for reading data from shared memory and reporting statistics that are accurate for the time during which the command executes. This is data that changes dynamically during processing such as buffers, locks and users.

Section 3.6 OS-Hardware Data

Every operating system has its own set of diagnostics files to track activity and failures. The most common are event logs and/or error reports. Here are a few:

AIX:	the /usr/bin/errpt –a command
HP-UX:	view the /var/adm/syslog/syslog.log file or run the /usr/bin/dmseg command.
Solaris:	view the /var/adm/messages file or run the /usr/bin/dmseg command. /usr/bin/truss to trace OS calls.
Linux:	view the /var/messages file.
Windows2000 & Windows NT:	system, security and application event logs also view windir\drwtsn32.log

On most UNIX systems, examination of /usr/include/sys/errno.h file will show error numbers (code) and their respective definition.

It is recommended that you refer to your OS documentation and supporting information to learn what other diagnostic information is available. It's also advisable to keep current of all operating system defects and available patches, this is easily accomplished by visiting the respective web-sites for periodic updates.

Section 3.6 Conclusion

There are various mechanisms available for collecting diagnostic data. A good place to start is examining the online message log for any information that may have been logged. The online message log will also reveal the location of any 'af.xxx' files that may have been created in response to an assertion failure. If a shared memory snapshot is created, this can be for forwarded to Informix Technical Support for analysis if needed. The IDS utility `finderr` is especially helpful with deciphering SQL & ISAM error codes and their respective meanings. Each OS has it own set of utilities to track activities and failures, please consult with your respective OS vendor to learn more about what utilities are available for your platform.

Section 4: A Case Example, Research & Informix Technical Support

Section 4.1 Introduction

Now that you know that a problem exists, and what elements are essential in resolving the problem, you can begin to research known problems, learn about workarounds and apply fixes. The majority of this section is dedicated on how to learn about known fixes, either externally or by contacting Informix Technical Support.

Section 4.2 A case example

The following is a case example to help you practice what you have learned in this tutorial.

The situation is as follows.
A certain user complains to you that the invoicing application repeatedly abends(fails) with the following SQL & ISAM errors:

> INF-240: Could not delete a row.
> ISAM-111: no record found

You begin discussing this situation with the user and determine:

The application that fails is the same program version that's been in production for over two years. The application creates shipping invoices for the next business day.
It's started occurring on Monday afternoon.
It reproduces EVERY time the application is run.
The client application that fails runs with Iconnect version 2.50
The database server version is 9.30.FC2
The OS version is AIX 4.3.3
If this application cannot be run, shipping will not be possible without invoices.

You investigate the meaning of the SQL & ISAM errors by using finderr:

```
$finderr 240
      -240    Could not delete a row.
      The database server cannot finish execution of a DELETE
      statement. Roll back the current transaction; then check the
      accompanying ISAM error
      code for more specific information.

$ finderr 11
      -111    ISAM error: no record found.

      The ISAM processor cannot locate the requested record. For C-ISAM
      programs, no record was found with the requested index value or
```



```
record number, depending on the retrieval mode in use. Make sure
that the correct index is in use. For SQL products, see the SQL
error message or return code. Probably no row was found for this
query.
```

Examination of the IDS online message log shows an assert failure message at the same timestamp the application failed:

```
08:50:02  Assert Failed: Fragid 0x1800003, Rowid 0x3d74c01
not found for delete
in partnum 1300000
8:50:02  Informix Dynamic Server Version 9.30.FC2
08:50:02   Who: Session(164, sapr3@fina03, 7667, 972141016)
               Thread(408, sqlexec, c000000039ef4a80, 3)
               File: exfmsupp.c Line: 826
08:50:02   Results: Delete failed
08:50:02   Action: Run 'oncheck –cI
prd:"sapr3".fmifiit#fmifiit~z2'
08:50:02  stack trace for pid 4523 written to
/tmp/af.580c58a
08:51:20   See Also: /tmp/af.580c58a
08:56:54  Checkpoint Completed:  duration was 4 seconds.
08:56:54  Checkpoint loguniq 453419, logpos 0x9bdb018
```

The message log failure message indicates the short description of what failed, in this situation "`Fragid 0x1800003, Rowid 0x3d74c01`", as well as the name of the table associated with the failure : `:"sapr3".fmifiit#fmifiit~z2".`

Also shown is the recommended Action: Run 'onhceck –cI' prd:sapr3:fmifiit#fmifiit~z2'

Analysis of the 'af.xxx file' (in this situation /tmp/af.580c58a) reveals the following information:

the "c" stack of the offending thread.
The current sql statement that was executing that is associated with the failing thread.

```
08:50:02  Assert Failed: Fragid 0x1800003, Rowid 0x3d74c01 not found
for delete in partnum 130000a
08:50:02   Who: Session(164, sapr3@fina03, 7667, 972141016)
               Thread(408, sqlexec, c000000039ef4a80, 3)
               File: exfmsupp.c Line: 826
08:50:02   Results: Delete failed
08:50:02   Action: Run 'oncheck -cI prd:"sapr3".fmifiit#fmifiit~z2'
08:50:02  Stack for thread: 408 sqlexec

 base: 0xc00000003eb52000
  len:   270336
   pc: 0x0000000000000000
  tos: 0xc00000003eb53ed0
state: running
```



```
   vp: 3

( 0)  0x40000000005075b0   legacy_hp_afstack + 0x250
[/informix/PRD/bin/oninit]
( 1)  0x4000000000506ae0   afstack + 0x78  [/informix/PRD/bin/oninit]
( 2)  0x4000000000505ebc   afhandler + 0x61c
[/informix/PRD/bin/oninit]
( 3)  0x40000000005057c0   affail_interface + 0x58
[/informix/PRD/bin/oninit]
( 4)  0x4000000000510e74   rptidxdelerr + 0x164
[/informix/PRD/bin/oninit]
( 5)  0x400000000058bc20   fm_idxdel + 0x2d8
[/informix/PRD/bin/oninit]
( 6)  0x4000000000580430   fmdelrec + 0x168  [/informix/PRD/bin/oninit]
( 7)  0x4000000000164084   sqisdelrec + 0x2c
[/informix/PRD/bin/oninit]
( 8)  0x400000000012ba2c   aud_sqisdelrec + 0x34
[/informix/PRD/bin/oninit]
( 9)  0x40000000001a89f8   dodelete + 0x218  [/informix/PRD/bin/oninit]
(10)  0x40000000001a7d48   chkrowcons + 0x9d8
[/informix/PRD/bin/oninit]
(11)  0x40000000001ac57c   dodmlrow + 0x76c  [/informix/PRD/bin/oninit]
(12)  0x40000000001abdd0   dodelupd + 0x738  [/informix/PRD/bin/oninit]
(13)  0x400000000012ffdc   aud_dodelupd + 0x14
[/informix/PRD/bin/oninit]
(14)  0x400000000024eb7c   excommand + 0x8d4
[/informix/PRD/bin/oninit]
(15)  0x40000000002224c4   sq_execute + 0xa4
[/informix/PRD/bin/oninit]
(16)  0x4000000000234618   sqmain + 0xf0  [/informix/PRD/bin/oninit]
(17)  0x40000000004e27d4   startup + 0xd4  [/informix/PRD/bin/oninit]
(18)  0x40000000004fbc1c   resume + 0x10c  [/informix/PRD/bin/oninit]

/informix/PRD/bin/onstat -g sql 164:

Informix Dynamic Server Version 9.30.FC2 -- On-Line -- Up 8 days
04:43:35 --
2170832 Kbytes

Sess  SQL            Current            Iso Lock       SQL  ISAM F.E.
Id    Stmt type      Database           Lvl Mode       ERR  ERR  Vers
164   DELETE         prd                DR  Wait       0    0    7.24

Current SQL statement :
  DELETE  FROM  "fmifiit"  WHERE  "mandt" =? AND  "fmbelnr" =? AND
    "fikrs" =? AND  "fmbuzei" =? AND  "btart" =? AND  "rldnr" =? AND
    "gjahr" =? AND  "stunr" =?

Last parsed SQL statement:
  DELETE  FROM  "resb"  WHERE  "mandt" =? AND  "rsnum" =?

The business impact is considered critical, as if this application
cannot be run, shipping will not be possible without generated
invoices.
```


At this point, we gathered the following information:

Problem and symptoms:	application fails with SQL error 240, ISAM error 111.
Business impact of problem:	critical: shipping halted.
Reproduction:	execution of invoicing application.
Information:	af.xxx file

Section 4.3 Case example - beginning analysis

In beginning our analysis of this case example, we ask ourselves the following questions:

1. Why, after two years of no reported problems with this application , we start to receive errors?

2. What does the information in the 'af.xxx' file tell me?

3. What could have changed that caused this problem to occur?

The answer to question #1 would be to investigate the application itself. Were there program revisions? Possible table structure changes?. In this case, no. But if there were, would those changes result in the types of SQL& ISAM errors received? Again, no. So, you can essentially eliminate application errors as root cause.

The answer to question # 2 is deciphering the information in the af.xxx file. This tells me that a DELETE operation on rowid xxx failed for table fmitiit. It also recommends an action, which is to run the oncheck utility on this table, with respect to the indexing option 'cI'. This hints to us that there is a possible problem with the indices on this table.

So, we run the `oncheck –cI prd:fmitiit` operation, and it reports no corruption or problems with the index. So, a reasonable conclusion can be made that the indices are not corrupted and not the root cause of the failure.

Moving on to question 3, we now begin to investigate any changes in the database or environment that were made recently. We ask the database and system administrators assigned to this instance if they are aware of any changes, such as maintenance (table reorganizations, chunk allocations, disk hardware changes, etc.) that were made recently. In this situation, we quickly learn that the table fmitiit was reorganized last weekend and the decision was made to fragment the table into multiple dbspaces.

So, at this point we learn that something did change, and that was the physical layout of the table on disk. It is no longer in the physical state that it was before the error started occurring. This is important information, as it implies that if the table can be returned to

its original state (non-fragmented) that this error may stop reproducing. This could then be considered an acceptable workaround. But we still yet have to learn why this error is occurring and if applying such a workaround would correct the situation.

It's recommended to contact the Informix Technical Support Department in order to report this problem ; the support engineer will most likely request the af.xxx file be made available for analysis. The engineer will discuss any potential bugs that may be responsible for root cause, and existing workarounds that are applicable.

Section 4.4 Newsgroups

Accessing newsgroups such as comp.databases.informix provides you with access to other IDS end users who can provide feedback and solutions t problems that they already encountered.

Section 4.5 Conclusion

In this section you have learned by working thru a case example. Specifically, how to describe a problem, determine its business impact, reproduction criteria, capture essential diagnostic data contacting Informix Technical Support department, and contemplate the application of a potential workaround.

Section 5: Summary

Section 5.1 What you should have learned

You now know how too properly identity, diagnose and provide solutions to Informix product defects. You have a basic understanding of Informix diagnostic tools and debug files, and now posses the knowledge to further increase your problem solving skills.